修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院情報理工学研究科情報・通信工学専攻　博士前期課程		
氏　　名	平井 芳孝	学籍番号	1331087
論 文 題 目	コンピュータ大貧民に対するニューラルネットワークと強化学習の適用		

要　　旨

本研究では，（大）富豪と（大）貧民間のカード交換後の手札およびカード提出後のゲーム状態から，大富豪から大貧民までの順位予測を試みた．カード交換後の手札から順位予測する実験では，学習パターン数を 50 万まで増やしたが正答率は 42¥%を越え，なお上昇し続けた．一方，カード提出後の事後状態から順位予測する実験では，学習パターン数を 50 万まで増やしたが正答率は 48¥%を越え，なお上昇し続けた．各特徴の影響を順位予測の正答率で確認する実験では，順位予測に影響を与えるゲーム状態の特徴は影響力の強いものから順に，カード残り枚数，相手の手札，自分の手札，使われたカード，革命・しばりの有無，場に出ているカードであった．予測順位に最も強い影響を与えると考えられるカード残り枚数について，自分のカード残り枚数が少なく相手のカード残り枚数が多いときに予測順位が高くなることを確認した．特に，自分のカード残り枚数が 0 枚で他プレイヤのカード残り枚数が 0 枚でないときには 1 位を正しく予測した．また，本研究の事後状態で使用した全特徴を使用した場合でも，自分のカード残り枚数が 0 のときには，上がりのプレイヤの人数に応じて，正しい階級を予測できた．

学習パターン数を増やすと，正答率が向上することを確認したので，TD-Gammon のように，膨大な回数の自己対戦から強化学習するコンピュータ大貧民プレイヤの開発を目指したい．

平成 26 年度 修士学位論文

コンピュータ大貧民に対する
ニューラルネットワークと強化学習の適用

1331087 平井 芳孝

指導教員　保木 邦仁

2015 年 2 月 16 日

電気通信大学大学院 情報理工学研究科
情報・通信工学専攻 情報数理工学コース

目 次

表 目 次

図 目 次

1　はじめに

本章では，本研究の研究背景と研究目的についてそれぞれ説明する．

1.1　研究背景

人工知能の研究では知能の理解だけでなく知能の構築も取り扱う．現在の人工知能分野では，学習や推論といった一般的な問題から，チェスや定理証明といった特定の問題にいたるまで，広大な領域を扱っている [5]．とりわけ，ゲームはルールが明確で，勝ち負けにより良し悪しを判断できるため，人工知能の性能評価に用いられてきた．そのうえ，ゲームによっては強い人間プレイヤがいるため，人知を超えるという目標も立てられる．

完全情報ゲームではゲーム木探索や評価関数の自動生成といった有効な手法が知られており，それらを適用した AI プレイヤはトップレベルの人間プレイヤの強さを持っている．オセロでは 1997 年に Logistello が世界チャンピオンの村上健に 6 戦 6 勝で勝利し [1]，チェスでは 1997 年に Deep Blue が世界チャンピオンの Garry Kimovich Kasparov に 2 勝 1 敗 3 引で勝利した [2]．将棋では 2014 年にドワンゴ・日本将棋連盟主催第 3 回将棋電王戦においてコンピュータがプロ棋士に対して 4 勝 1 敗で勝利し [3]，囲碁では 2014 年に第 2 回電聖戦において Crazy Stone が 19 路盤 4 子局で依田紀基九段に対して 2 目半勝ちした [4]．

不完全情報ゲームにおいては，完全情報ゲームほど研究が進んでいない．不完全情報ゲームはプレイヤごとに得られるゲームの状態に関する情報が部分的である．そのため，探索を開始する初期のゲーム状態を完全に規定するには，可能性のあるゲーム状態を考慮せねばならず，探索するゲームの状態数が爆発的に増加する．よって，完全情報ゲームで有効とされていた手法を直接適用できない．

不完全情報ゲームの一種に大貧民がある．2009 年，モンテカルロ法とその制御に UCB1-TUNED を用いた AI プレイヤが第 4 回 UEC コンピュータ大貧民大会において優勝を収めた [6]．それ以降，モンテカルロ法はコンピュータ大貧民における強い AI プレイヤにとっての主流の手法となっている．一方，コンピュータ大貧民において評価関数の機械学習法に基づく AI プレイヤ生成の試みはあまり盛んではない．

1.2　研究目的

本研究では，機械学習法の一種である強化学習法の大貧民への適用を念頭に置き，事後状態から順位予測を試みる．強化学習法は不完全情報ゲームの一種であるバッグギャモンで，バッグギャモンについての予備知識をほとんど必要せずに，グランドマスターに近いレベルの手を指せる TD-Gammon というプログラムの開発に寄与した [8]．本研究の研究対象である大貧民もバッグギャモンと同様に不完全情報ゲームの一種である．大貧民とバッグギャモンはプレイ人数が違うものの，どちらも不完全情報ゲームである．このように共通した性質を持つため，バッグギャモンで成功した手法が大貧民においても成功する可能性がある．

2　基礎知識

本章では，本研究の実験内容を理解するために必要な基礎知識について説明する．まず，本研究の研究対象である大貧民というゲームのルールを説明する．そして，順位予測するために使用するニューラルネットワークとそれの重みや閾値を調整する誤差逆伝播法について説明する．さらに，強化学習の仕組みと事後状態について説明する．

2.1　大貧民

大貧民 (「大富豪」,「階級闘争」などとも呼ばれる) はトランプゲームの一種である．カードを参加プレイヤに対してできるだけ均等な枚数になるように配り，各プレイヤは手持ちのカードを場に出して，早く手札をなくす (上がる) ことを競う．

このゲームには数多くのローカルルールが存在する．ここでは，UEC コンピュータ大貧民大会で採用されているルールを説明する [9]．

- 基本的なルール
  - プレイ人数…5 人
  - カードの枚数…ゲームに使用するカードは各スート (スペード，ハート，ダイヤ，クラブ) のエースからキングまでの 52 枚にジョーカー 1 枚を加えた 53 枚である．
  - ランクの強さ…通常時のランクの強さは $3 < 4 < 5 < 6 < 7 < 8 < 9 < 10 < J < Q < K < A < 2$ であり，革命時のランクの強さは $2 < A < K < Q < J < 10 < 9 < 8 < 7 < 6 < 5 < 4 < 3$ である．ジョーカーは通常時でも革命時でも常に最強のカードとして扱われる．通常時であれば 2 より強く，革命時であれば 3 よりも強い．通常時と革命時については「場の状態」の項目にて後述する．
  - 階級…初回のゲームは全プレイヤが平民の階級である．二回目以降のゲームでは，1 つ前のゲームにおいて上がったプレイヤから順に大富豪，富豪，平民，貧民，大貧民という階級付けを行う．また，階級に応じて二回目以降のゲーム開始前にプレイヤ間でカードの交換が行われる．それぞれの階級がどのように手札を交換するのかを表 1 に示す．

    貧民と大貧民は交換に出すカードを自由に選択できない．一方，大富豪と富豪は自分の手札から交換に出すカードを自由に選択できる．なお，貧民と大貧民の場合で同じランクのカードが複数ある場合には，次の優先順位にしたがって交換に出すカードを決定する．

    1. スペード
    2. ハート
    3. ダイヤ
    4. クラブ

表 1: 階級とカード交換

順位	階級名	カード交換で行う作業
1	大富豪	大貧民に任意のカードを 2 枚渡す
2	富豪	貧民に任意のカードを 1 枚渡す
3	平民	なにもしない
4	貧民	富豪に最も強いカードを 1 枚渡す
5	大貧民	大富豪に強いカードを 2 枚渡す

  - カードの配布方法…初回のゲームは席順 (カードを出す順番) をランダムに決定し，カードを配布する．階級が決定した二回目以降のゲームは大富豪を起点として席順にカードを配布する．
  - 上がり…自分に配布されたカードをすべて場に出すこと．
  - ポイントの割り当て方…大富豪は 5 点，富豪は 4 点，平民は 3 点，貧民は 2 点，大貧民は 1 点を得る．
  - 対戦方法…5 プレイヤを 1 つのグループとし，100 回の対戦を行う．
  - 勝敗の決定方法…100 回の対戦で獲得した合計ポイントが最も多いプレイヤの勝利となる．

- 提出手役

  提出手役には単独出し，複数出し，階段がある．

  - 単独出し ‥ 1 枚のカードを場に出す提出手役．ただし，すでにカードが場に出ている場合は，場に出すカードが場に出ているカードよりも強くなければならない．
  - 複数出し…複数枚の同じランクのカードを場に同時に出す提出手役．ただし，すでに同じランクの複数枚のカードが場に出ている場合，場に出すことができるカード（提出カード）の条件は以下の 2 点である．
    * 提出カードが場に出ているカードよりも強いこと
    * 提出カードが場に出ている，複数枚の同じランクのカードと同じ枚数によって構成されていること
  - 階段…同じスートでランクが 3 以上連続したカードを場に同時に出す提出手役．ただし，すでに階段が場に出ている場合，場に出すことができるカード（提出カード）の条件は以下の 2 点である．
    * 提出カードのそれぞれが場に出ているカードよりも強いこと
    * 提出カードが場に出ている階段と同じ枚数で構成された階段であること

- 場の状態

- 革命…革命が発生すると，カードの強弱が逆転する．革命は次の条件のいずれかが満たされたときに発生する．ただし，革命を起こすかどうかをプレイヤは選択できない (次の条件を満たすと革命が必ず発生する)．なお，革命時でないときは通常時と呼ぶ．革命時に革命が発生すると通常時になる．
  - ＊ 同じランクのカード 4 枚が場に同時に出されたとき
  - ＊ 同じランクのカード 3 枚とジョーカーが場に同時に出されたとき
  - ＊ 同じランクのカード 4 枚とジョーカーが場に同時に出されたとき
  - ＊ 5 枚以上のカードが階段として場に出されたとき
- しばり…同じスートのカードが連続した手番で場に出された場合，しばりが発生する．しばりが発生すると場に出ているスートと同じスートのカードしか場に出せない．複数出しや階段においても出すカードのスートすべてが場に出てるカードのスートすべてと一致していれば，しばりは発生する．

- 特殊なカードの効果
  - スペードの 3…ジョーカーが単独で場に出されている場合，スペードの 3 をジョーカーよりも強いカードとして場に出すことができる．そして，場は流れる．
  - 8 切り…8 のカードが場に出されると場は流れる．8 を含んだ複数出しや階段でも場は流れる．
  - ジョーカー…ジョーカーはどんなカードとしても使うことができる． たとえばランク 9 のカードを 2 枚持っているとき，ジョーカーを加えて 9 の 3 枚組として場に出すことができる． ジョーカーを含んだ提出カードでしばりや革命を起こすことも可能である．しかしながら，ジョーカーを単独で使用する場合，スペードの 9 といったように特定のカードとすることはできない．単独で出されたジョーカーは，常に最強のカードとして扱われる．

- ゲームの進め方
  - ゲームの開始…カード交換後，各プレイヤの手札が決定したら，通常時の状態からゲームを開始する．ゲームはダイヤの 3 を持っている人から始まる．
  - 手番の回り方…席順 (カードを出す順番) は乱数で決定される．その席順で何ゲームか行った後，再度席替えが行われる．
  - 場…各プレイヤがカードを出す場所．場は全プレイヤ共通で一つしかない．
  - パス…パスとは自分の手番でカードを場に出さないこと．あるプレイヤが一度パスをすると，場が流れるまでそのプレイヤの手番はまわってこない．
  - カードの出し方…場に出ているカードがないときは任意の提出手役 1 つを出すことができる．カードが場に出ているときは，場に出ているカードよりも強いカードからなる提出手役 1 つを出すか，パスをする．

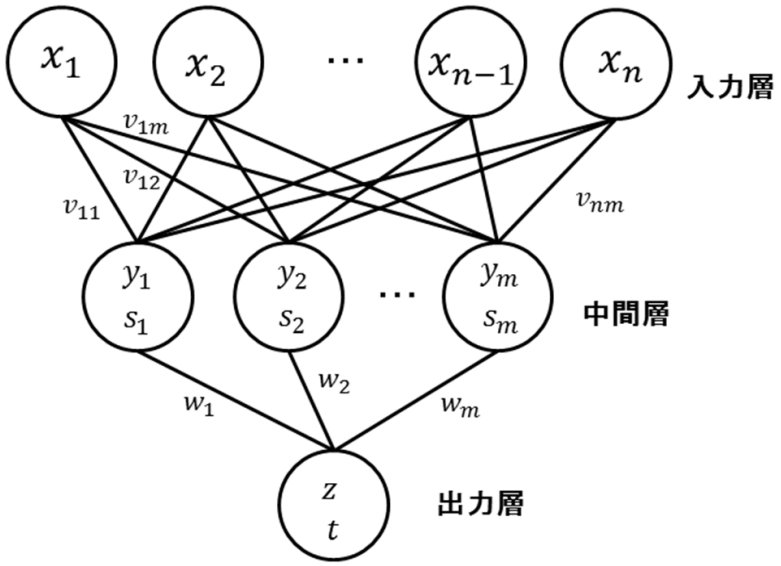

図 1: 三層ニューラルネットワーク

- 場の流れ方…すべてのプレイヤがパスをすると，場が流れる (プレイ人数が 5 人なので 5 回目のパスの後に，場が流れる)．場が流れることによって，いま場に出てるカードはすべて場の脇に置く．場が流れるとしばりの効果は消えるが，革命の効果は持続される．そして，カードを最後に出した人に次の手番が回る．もし自分以外のプレイヤが全員パスをしていて，自分がまだ出せるカードがある場合，自分に手番が何度も回ってくる．
- 千日手…すべてのプレイヤが延々とパスをし続けた場合，次のような処理を行い順位を決定する．
  - すでに上がったプレイヤは通常通りの順位がつく．
  - パスが 20 回続いた場合，まだ上がっていないプレイヤの順位はランダムに決定される．

2.2 ニューラルネットワーク

ニューラルネットワークとは脳の神経を数学的に記述することを目指して作られた数学的モデルの一種である [11]．図 1 に三層ニューラルネットワークの構成を示す．三層ニューラルネットワークは入力層，中間層，出力層の 3 つの層からなる．さらに，入力層には n 個のユニット，中間層には m 個のユニット，出力層には 1 個のユニットがあるとする．x_1 から x_n は入力層の各ユニットの値であり，y_1 から y_m は中間層の各ユニットの値であり，z は出力層のユニットの値である．s_1 から s_m は中間層の各ユニットの閾値であり，t は出力層のユニットの閾値である．v_{ij} は入力層の i 番目のユニットと中間層の j 番目のユニットの結合の重みであり，w_j は中間層の j 番目のユニットと出力層のユニットの結合の重

みである．なお，この 2.2 節における記号の表記法は次の 2.3 節における誤差逆伝播法の説明で使用する．

2.3 誤差逆伝播法

誤差逆伝播法 (バックプロパゲーション) はニューラルネットワークの重みや閾値を調整する教師付き学習アルゴリズムの一種である [11]．誤差逆伝播法で学習を行うには学習パターンを用意する必要がある．ここでは，学習パターンを N 個用意したとする．学習パターンは入力とそれに対応するべき出力の組からなる．入力の値はニューラルネットワークの入力層に与える値 $\boldsymbol{x}$ であり，出力の値は入力に対する理想の値 z_d である．

次に，誤差逆伝播法の処理内容について説明する．誤差逆伝播法では，Step 1 を一回処理したあとに，N 個の学習パターンそれぞれに対して Step 2 から Step 14 までの処理をする．本論文では，N 個の学習パターンそれぞれに対して Step 2 から Step 14 まで処理することを一回の学習と表現する．学習の回数は実験者が求める平均二乗誤差の値や任意の回数などで指定する．Step 2 から Step 14 は平均二乗誤差を小さくする．平均二乗誤差を式 (1) に示す．

$$\frac{\sum_{k=1}^{N}(F(\boldsymbol{x})-z_d)^2}{N} \tag{1}$$

平均二乗誤差は N 個の学習パターンそれぞれに対して学習パターンの出力 z_d と，学習パターンの入力 $\boldsymbol{x}$ をニューラルネットワークに与えたときのニューラルネットワークの出力 $F(\boldsymbol{x})$ の二乗誤差の総和を学習パターン数 N で割ったものである．平均二乗誤差が小さくなることは，N 個の学習パターンにわたって，z_d と $F(\boldsymbol{x})$ が接近することを意味し，それを本論文で学習がうまくいっていると解釈する．

Step 1 における学習係数は値が大きいと Step 7 や Step 10 における一回の更新での重みの値の変化が大きくできるが，平均二乗誤差も大きくなる．また，慣性係数は，値を適切に設定すると重みや閾値の振動が抑制され，学習の効率がよくなる．学習係数や慣性係数の最適な値を決定する一般的な方法は知られておらず，色々な値を設定し実験することでよさそうな値を決める．

なお，以下の式 (3) と (4) における関数 f は活性化関数と呼ばれており，式 (2) で示す標準シグモイド関数がよく使われる．

$$f(x)=\frac{1}{1+e^{-x}} \tag{2}$$

Step 1 ユニットの閾値 s_j と t，結合の重み v_{ij} と w_j を乱数で初期化する．学習係数 η と慣性係数 α には適当な値を与える．

Step 2 学習パターンの入力をニューラルネットワークの入力層の各ユニットに与える．

Step 3 m 個ある中間層の j 番目のユニットの値を以下の式 (3) にしたがって求める．

$$y_j \leftarrow f(\sum_{i=1}^{n} x_i v_{ij} + s_j) \tag{3}$$

Step 4 出力層のユニットの値を以下の式 (4) にしたがって求める．

$$z \leftarrow f(\sum_{j=1}^{m} y_j w_j + t) \tag{4}$$

このようにして，学習パターンの入力をニューラルネットワークの入力層に与え計算して求めた出力を本章では計算値として z_c と表記する．

Step 5 理想の値 z_d と計算値 z_c を用いて，重み w_j を更新するのに使う，出力層のユニットの学習信号 σ を式 (5) で求める．なお，$z_d - z_c$ は誤差で，微分係数 $z_c(1 - z_c)$ はシグモイド関数の導関数の値である．つまり，式 (5) は誤差と微分係数の積を意味している．

$$\sigma \leftarrow z_c(1 - z_c)(z_d - z_c) \tag{5}$$

Step 6 中間層の j 番目のユニットと出力層のユニットの間における重みの修正量 Δw_j を式 (6) で求める．

$$\Delta w_j \leftarrow \eta \sigma y_j + \alpha \Delta w_j \tag{6}$$

Step 7 中間層の j 番目のユニットと出力層のユニットの間における重み w_j を式 (7) で修正する．

$$w_j \leftarrow w_j + \Delta w_j \tag{7}$$

Step 8 重み v_{ij} を更新するのに使う，中間層の j 番目のユニットにおける学習信号 τ_j を式 (8) で求める．

$$\tau_j \leftarrow y_j(1 - y_j)\sigma w_j \tag{8}$$

Step 9 入力層の i 番目のユニットと中間層の j 番目のユニットにおける重みの修正

量 Δv_{ij} を式 (9) で求める．

$$\Delta v_{ij} \leftarrow \eta \tau_j x_i + \alpha \Delta v_{ij} \tag{9}$$

Step 10 入力層の i 番目のユニットと中間層の j 番目のユニットにおける重み v_{ij} を式 (10) で修正する．

$$v_{ij} \leftarrow v_{ij} + \Delta v_{ij} \tag{10}$$

Step 11 出力層のユニットにおける閾値の修正量 Δt を式 (11) で求める．

$$\Delta t \leftarrow \eta \sigma + \alpha \Delta t \tag{11}$$

Step 12 出力層のユニットにおける閾値 t を式 (12) で修正する．

$$t \leftarrow t + \Delta t \tag{12}$$

Step 13 中間層の j 番目におけるユニットの閾値の修正量 Δs_j を式 (13) で求める．

$$\Delta s_j \leftarrow \eta \tau_j + \alpha \Delta s_j \tag{13}$$

Step 14 中間層の j 番目におけるユニットの閾値 s_j を式 (14) で修正する．

$$s_j \leftarrow s_j + \Delta s_j \tag{14}$$

以上の手順を繰り返すと，ニューラルネットワークに学習パターンの入力を与えて求めた計算値 z_c と学習パターンの出力である z_d の誤差の二乗が小さくなり，ある入力に対して，その入力に対応すべき出力をするような関数を近似的に得ることができる．

また，誤差逆伝播法で学習したニューラルネットワークの性能を確認するためにテストパターンが必要である．学習パターンとテストパターンは名称が違うものの，パターンを学習に使うのかテストに使うのかの違いしかなく，データ形式上は同じものである．ただ，一般的に，ニューラルネットワークは未知のパターンに対しての性能が求められるので，学習パターンとテストパターンには異なるものを用いることが多い．

次に，テストパターンの使い方を説明する．まず，上のアルゴリズムにおける Step 2 で，学習パターンではなくテストパターンを与える．そして，Step 3 から Step 4 までの処理をし，テストパターンの計算値を求める．なお，Step 2 から Step 4 までの処理過程はフォワードプロパゲーションと呼ばれている．それから，その計算値と理想の値 (テストパターンの出力) を比較する．計算値と理想の値が近いほど，ニューラルネットワークが未知の関数をより良く近似している．

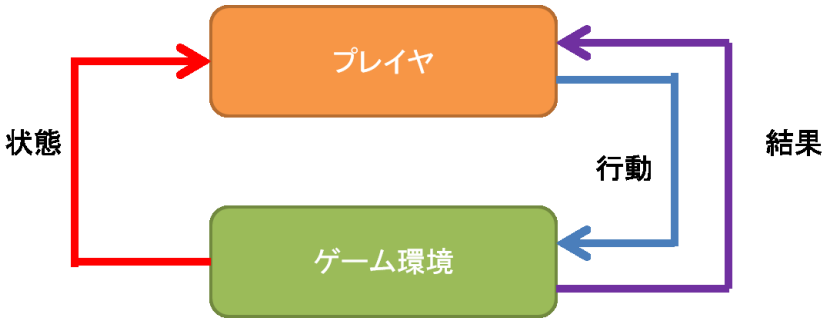

図 2: 強化学習概念図

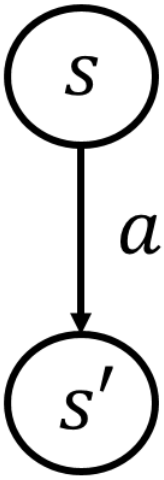

図 3: 事後状態概念図

2.4 強化学習

本研究は強化学習法へ応用することが目的であるが，強化学習法自体はまだ使用していない．使用しているのは強化学習法へ応用するための事後状態という概念である．よって，強化学習自体については概要を示し，本研究を理解するために必要な概念である事後状態について説明する．

図 2 にゲーム環境における強化学習概念図を示す．ある状態においてプレイヤが行動すると，その行動に対してゲーム環境が結果を返す．そして結果に応じて，プレイヤはよりよい結果が得られるように行動を変える．

図 3 に事後状態概念図を示す．事後状態 s' とはある状態 s で行動 a を選択した後の状態である．とりわけゲームにおける事後状態では，自分の手番ではなく相手の手番になっていなければならない．

次に，事後状態を用いることの利点を三目並べの例を用いて図 4 に示す．水色の ◯× を含む盤面図は状態を表し，オレンジ色の ◯× を含む局面は行動を表している．従来の行

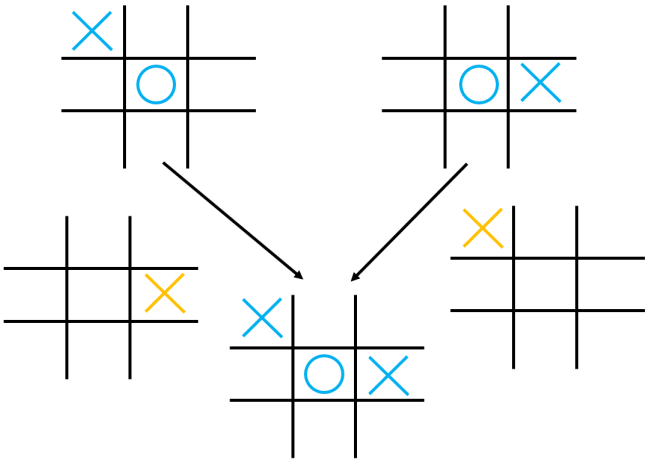

図 4: 合流する事後状態

動を評価する関数では図 4 における右側の指し手と左側の指し手の価値は異なる．しかしながら，右側の指し手も左側の指し手も同じ事後状態を生成するので，本来は同じ価値であるのが望ましい．従来の行動を評価する関数は状態と指し手を別々に評価するが，事後状態を評価する関数では状態と指し手をまとめてひとつの事後状態として一度に評価できる．また，事後状態がどのような状態でどのような行動によって発生したのかを知らなくても評価関数の作成に利用することができる．

3　従来研究

本研究と関連の深い 2 つの研究を紹介する．1 つ目は地曳と松崎の “大貧民における棋譜データからの提出手役評価関数の学習” である [7]．本研究の大貧民に機械学習法を適用するアイデアはこの研究から得られた．2 つ目は Tesauro の “Temporal Difference Learning and TD-Gammon” である [10]．本研究の事後状態から順位予測するアイデアはこの研究から得られた．なお，TD-Gammon が対象とするバックギャモンというゲームは二人ゲームなので勝敗予測だが，本研究が対象とする大貧民は 5 人ゲームなので，予測するのは勝敗ではなく順位である．以下でそれぞれの研究について説明する．

3.1　大貧民に機械学習を適用する研究

地曳と松崎は「棋譜データにおける提出カード」と「提出カード評価関数によって得られた評価値が最も大きい提出カード」の一致率を報告した．

学習パターンを作成するために 2 種類の大貧民の棋譜データを利用した．1 つ目は各合法手に対して 500 回のプレイアウトを行う原始モンテカルロ法プレイヤ 5 名の対戦を記録したものである．もう 1 つは人間のプレイヤ 1 名と各合法手に対して 500 回のプレイアウトを行う原始モンテカルロ法プレイヤ 4 名の対戦を記録したものである．

ニューラルネットワークには以下の特徴を入力する．

- 場に関する情報
  - 革命の有無
  - しばりがある場合にはそのスート
  - 最後に場に出された提出カード
  - 各プレイヤがその手番でプレイ可能かどうか
- カードに関する情報
  - 各プレイヤが持つカードの枚数
  - 場に出てるカード
  - 次にプレイするプレイヤが持つカード
- 提出カードに関する情報
  - 次にプレイするプレイヤが実際に出した手役

カード提出する前のゲーム状態を p，棋譜データで採用された合法手を a，棋譜データで採用されなかった k 番目の合法手を a'_k，学習パターンの入力から 2.3 節の計算値 z_c を求める関数を F，2.3 節の式 (2) と同じシグモイド関数を S，不採用の合法手数を M としたとき，以下の式 (15) で示される目的関数を最小化するように誤差逆伝播法で三層ニューラルネットワークを学習させた．この目的関数を最小化すると，棋譜データで採用された提出カードの採用度を上げ，棋譜データで採用されなかった提出カードの採用度を下げる．その結果，棋譜データと同じような提出カードを合法手として選択できる評価関数が生成される．

$$\sum_{k=1}^{M} S(F(p, a'_k) - F(p, a)) \tag{15}$$

提出カード評価関数の学習には三層ニューラルネットワークと誤差逆伝播法を使用した．三層ニューラルネットワークの入力層におけるユニット数は 120，中間層におけるユニット数は 15 と 50 の 2 通り，出力層におけるユニット数は 1 である．重みと閾値は初期値が異なるものを 100 個用意した．中間層のユニット数が 15 の場合は学習パターン数を 1000 から 19000 まで 2000 刻みで用意した．中間層のユニット数が 50 の場合は学習パターン数を 1000 から 27000 まで 2000 刻みで用意した．すべての学習パターンに対して重みや閾値を 1 回だけ調整することを 1 回の学習とし，1000 回の学習を行った．学習係数は 0.9 を初期値とし，1 回の学習ごとに 0.99 を掛けて使用した．なお，慣性係数は不明である．提出カード評価関数の性能を調査するため，学習時に使用していないパターンを 1000 個用意した．

実験の結果，学習に使用する学習パターン数を増やすことで提出カードの一致率は上昇した．15000 個の学習パターンで学習した提出カード評価関数では，学習に使用していない未知のゲーム状態に対する提出カード一致率がおよそ 69%となった．この結果は，大貧民といったある程度複雑なゲームにおいても，三層ニューラルネットワークが有効だと示している可能性がある．

3.2 事後状態から勝敗予測する研究

Tesauro は バックギャモンを学習するプログラム TD-Gammon の開発を目的とした研究の成果を報告した．

TD-Gammon とはバックギャモンで自分自身と対戦し，その結果から学習することで，バックギャモンの局面評価関数となるようなニューラルネットワークを用いた AI プレイヤである．

TD-Gammon は地曳と松崎における研究や本研究と同様に三層ニューラルネットワークを用いており，入力層のユニット数は 198，中間層のユニット数は 40 から 160 (TD-Gammon のバージョンが上がるにつれて増量)，出力層のユニット数は 1 である．

TD-Gammon の学習には地曳と松崎における研究や本研究と同様に誤差逆伝播法を使用している．よって，TD-Gammon を学習させるためには学習パターンが必要である．学習パターンの入力はあるゲーム状態に対してある合法手を選択した後の事後状態で，学習パターンの出力はその事後状態からの勝敗である．なお，TD-Gammon の学習は 1 回のゲームが終了するたびに行われる．

学習パターンを生成するために TD-Gammon に自己対戦を行わせた．自己対戦の開始時において，TD-Gammon の重みや閾値はランダムな小さい値に設定された．したがって，自己対戦の初期において，TD-Gammon が計算した評価値 (勝率) はランダムな値である．この評価値を基準に合法手を選択するので，自己対戦の初期は弱い手しか指せない．しかしながら，何十回かゲームをすると，性能は急速に向上した．

自己対戦を 300,000 回行った TD-Gammon 0.0 は当時最強のバックギャモン AI プレイヤと互角に対戦できるほど強化された．当時最強のバックギャモン AI プレイヤはバックギャモンについての膨大な知識を利用していたので，バックギャモンについての知識を与えずに作られた TD-Gammon がここまで強くなったのは驚くべき結果であった．

そして，TD-Gammon 0.0 にバックギャモン固有の知識を導入して，TD-Gammon 1.0 が開発された．この TD-Gammon 1.0 は従来のバックギャモン AI プレイヤすべてに対して圧倒的な強さを見せ，人間のエキスパートプレイヤと互角に戦うことができた．

TD-Gammon 2.0 では選択的 2 段階探索が導入された．選択的 2 段階探索とは選んだ手の直後のゲーム状態だけでなく，相手の出すサイコロの目と，それぞれに対応する合法手についても先読みを行うことである．その際，相手は最善手を選択すると仮定する．なお，計算時間を節約するために高く評価された 4 手から 5 手のみについて選択的 2 段階探索をおこなった．結果，TD-Gammon 2.0 はグランドマスターレベルに至った．

TD-Gammon 3.0 では中間層のユニットを 160 個使い，選択的 3 段階探索が導入された．

TD-Gammon 3.0 は世界最高の人間プレイヤと同程度の強さを持っていると思われおり，すでに世界チャンピオンとなっている可能性もある．

TD-Gammon の登場によって，世界最高クラスの人間プレイヤも知らなかった定石が発見され，TD-Gammon はバックギャモンの文化に貢献した．

4 実験

本研究の最終目標は強化学習法を大貧民に適用し，強い AI プレイヤを実現することである．そのため，カード交換後の手札やカード提出後のゲーム状態から順位予測できるかどうか，および学習パターン数を増やしたときに順位予測の正答率が向上するかどうかをそれぞれ確かめる．

まず，実験に必要な準備について説明する．それは，棋譜データの作成，学習やテストに使用するパターンの表現方法および作成方法，ニューラルネットワークの構造，誤差逆伝播法の設定である．そして，本研究の実験内容について説明する．それは，カード交換後の手札からの順位予測，カード提出後の事後状態からの順位予測，各特徴の影響調査，カード残り枚数と順位の関係である．

なお，基礎知識の誤差伝播法における計算値を本章では評価値と呼ぶ．

4.1 棋譜データの作成方法

棋譜データはコンピュータ大貧民の AI プレイヤを自動対戦させて作成した．その際に UEC コンピュータ大貧民大会のサーバを利用した [12]．学習パターンを増やしたときに性能がよくなることを確認したいため，棋譜データはある程度強い AI プレイヤを利用して作成する必要がある．そこで，棋譜データの作成には第五回 UEC コンピュータ大貧民大会の優勝プログラム snowl を利用した [12]．AI プログラム snowl は，モンテカルロ法とその制御に UECB1-TUNED をコンピュータ大貧民に初めて適用した，第四回 UEC コンピュータ大貧民大会の優勝プログラムを強化したものである．この snowl を 5 つ対戦させ，サーバとクライアントの通信を記録することで棋譜データを作成した．棋譜データには，プレイヤ番号，カード交換前の手札，交換に出すカード，各手番における自分の手札，場に出すカード，場に出てるカード，各プレイヤの手札のカード枚数，各プレイヤの階級，場が流れたか否か，革命が発生しているか否か，しばりが発生しているか否かなどを記録した．棋譜データは約 14 万 (139,151) のゲーム数，約 1200 万 (11,571,707) の手番数からなる．なお，本研究における実験で作成した学習パターンはすべてこの棋譜データを使用した．

4.2 学習パターンやテストパターンの表現方法

本研究では，大きく分けて 2 種類のパターンを作成した．なお，出力はどちらも予測順位にもとづく有利不利の評価値なので，2 種類のパターンの入力をそれぞれ説明した後に，

まとめて説明する．

1つ目のパターンはカード交換後実験で使用するパターンである．カード交換後実験で使用するパターン (カード交換後パターン) の入力について説明する．ゲーム開始前に，大富豪の場合は大貧民に渡すカードを2枚，富豪の場合は貧民に渡すカードを1枚それぞれ選択しなければならない．しかしながら，大富豪，富豪が渡したカードではなく，カードを渡した後に残っている最初の手札に注目することにした．そうすることで，大富豪・富豪のカード交換後の手札だけでなくカード交換後における全プレイヤの最初の手札を学習パターンの入力に使用することができる．なお，カード交換前の手札とカード交換後の手札を比較することで，どのカードが交換に出されたのかを知ることができる．パターンの入力は♠3，♠4，…，♠K，♠A，♠2，♡3，♡4，…，♡K，♡A，♡2，♢3，♢4，…，♢K，♢A，♢2，♣3，♣4，…，♣K，♣A，♣2，Joker で，各カードを持っている場合は1，持っていない場合は0とした．

2つ目のパターンはカード提出後実験で使用するパターンである．カード提出後実験で使用するパターン (カード提出後パターン) の入力について説明する．各プレイヤは自分の手番が回ってきたら，パスまたはカード提出をしなければならない．パターンの入力となる事後状態は以下の特徴から構成した．いずれも自分の行動選択後の情報となる．

- 自分の手札 (53 ユニット)
  53 枚のカードの有無を表す．各カードについて，自分の手札にある場合は1，ない場合は0とした．

- 相手の手札 (53 ユニット)
  53 枚のカードの有無を表す．各カードについて，他プレイヤ 4 人のいずれかの手札にある場合は1，ない場合は0とした．

- 使用されたカード (53 ユニット)
  53 枚のカードの有無を表す．各カードについて，全プレイヤのいずれかの手札にもない場合は1，ある場合は0とした．

- 場に出てるカード (53 ユニット)
  53 枚のカードの有無を表す．各カードについて，場の一番上にある場合は1，ない場合は0とした．もし自分がなんらかのカードを場に出した場合は，場に出てるカードとは自分の出したカードである．一方，自分がパスを選択した場合は，自分よりも前のプレイヤが場に出したカードがそのまま使われる．

- 各プレイヤのカード残り枚数 (12 ユニット ×5)
  各プレイヤのカード残り枚数を表す．UEC コンピュータ大貧民大会では5人対戦なので，5人の最初の手札の枚数はそれぞれ11枚，11枚，11枚，10枚，10枚であり，最大で11枚である．よって，手札の枚数は0枚から11枚の12通りが存在する．そこで，手札の枚数を以下のようなビット列で表現した．

表 2: カード枚数とビット列

0 枚	000000000001
1 枚	000000000010
2 枚	000000000100
3 枚	000000001000
4 枚	000000010000
5 枚	000000100000
6 枚	000001000000
7 枚	000010000000
8 枚	000100000000
9 枚	001000000000
10 枚	010000000000
11 枚	100000000000

- 革命中かどうか (1 ユニット)
  革命中であれば 1，そうでなければ 0 とした．

- しばり中かどうか (1 ユニット)
  しばり中であれば 1，そうでなければ 0 とした．

パターンの出力は予測順位にもとづく評価値である．ゲームからプレイヤが上がるたびにプレイヤの階級が決定するので，それを利用した．評価値は，大富豪の場合は 0.9，富豪の場合は 0.7，平民の場合は 0.5，貧民の場合は 0.3，大貧民の場合は 0.1 とした．このように設定したのは，パターンをニューラルネットワークに入力し，フォワードプロパゲーションをすると，最終的に出力するのはシグモイド関数の影響で 0 から 1 の浮動小数点数になり，それと合わせるためである．このように設定することで，フォワードプロパゲーションの結果である，予測順位にもとづく評価値が 0 以上 0.2 未満ならば大貧民，0.2 以上 0.4 未満ならば貧民，0.4 以上 0.6 未満ならば平民，0.6 以上 0.8 未満ならば富豪，0.8 以上 1.0 未満ならば大富豪と判断することができる．

4.3 学習パターンやテストパターンの作成方法

まず，カード交換後パターンの作り方について説明する．最初，棋譜データを読む込む．そして，カード交換後の最初の手札を記録する．なお，平民の場合は，最初の手札をそのまま記録する．このカード交換後の最初の手札がカード交換後パターンにおける入力となる．そして，プレイヤが上がると，そのプレイヤの順位を記録する．この順位がカード交換後パターンにおける出力となる．よって，カード交換後パターンは 1 ゲーム 1 プレイヤで 1 個だけ作成できる．

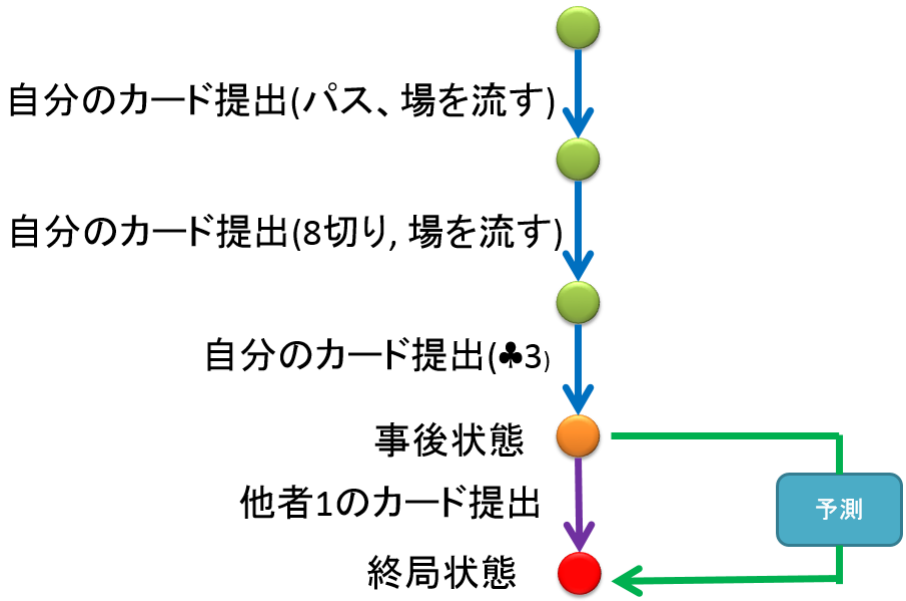

図 5: 1 プレイヤの連続した行動選択

つぎに，カード提出後パターンの作り方について説明する．最初，棋譜データを読み込む．そして，プレイヤ P の手番でプレイヤ P の連続した行動のうち，最後の行動後のゲーム状態 (事後状態) を記録する．この事後状態がカード提出後パターンの入力となる．そして，プレイヤが上がると，そのプレイヤの順位を記録する．この順位がカード提出後パターンにおける出力となる．1 ゲーム中の事後状態すべてに順位を対応づけるので，カード提出後パターンは 1 ゲーム 1 プレイヤで事後状態の個数だけ作成できる．

なお，大貧民における事後状態の認識には注意しなければならない点があるので，その例を図 5 に示す．

チェスや将棋といったゲームでは自分の手番と相手の手番が交互に変わるので，自分の行動後の状態は必ず事後状態となる．しかしながら，多人数ゲームの一種である大貧民ではそうとはかぎらない．自分の手番の後に再び自分の手番となる場合が 2 つある．1 つ目は 5 回目のパスをした場合である．UEC コンピュータ大貧民大会における大貧民のルールは，一般的な大貧民のルールと違い，自分以外の全プレイヤがパスをしたときではなく，自分も含めた全プレイヤがパスをしたときに場が流れる．これは自分が出したカードに自分自身がさらにカードを出せるルールになっているためである．ここから，5 回目のパスをするプレイヤは最後にカードを場に出したプレイヤということがわかる．なので，5 回目のパスの後は場が流れ，再び自分の手番になる．よって，5 回目のパスの後は事後状態としてはならない．2 つ目は 8 切りが発生した場合である．8 切りが発生すると，ほかのプレイヤに手番が移らずに場が流れる．なので，8 切りが発生すると再び自分の手番になる．よって，8 切り発生後は事後状態としてはならない．

このような理由で，本研究では 5 回目のパスをした後と 8 切りが発生した後の状態は事後状態に含めなかった．

4.4　ニューラルネットワークの構造

本研究におけるすべての実験において，三層ニューラルネットワークを利用した．カード交換後の手札から順位予測する実験では，入力層のユニット数を 53，中間層のユニット数を 25，出力層のユニット数を 1，重みの初期値を -0.3 から 0.3 までの乱数，閾値の初期値を 0 とした．カード提出後の事後状態から順位予測する実験，各特徴の影響を順位予測の正答率で確認する実験およびカード残り枚数で順位予測する実験では，入力層のユニット数を 274，中間層のユニット数を 100，出力層のユニット数を 1，重みの初期値を -0.3 から 0.3 までの乱数，閾値の初期値を 0 とした．

4.5　誤差逆伝播法における設定

本研究におけるすべての実験で，活性化関数を標準シグモイド関数，慣性係数を 0.8，学習係数を 0.01 とした．

4.6　カード交換後の手札から順位予測する実験

順位 R_d で上がることができた，カード交換後の手札から順位予測を行い，予測した順位 R_c と R_d の一致率を調査した．なお，一致率は後述する正答率，1 ズレ率，2 ズレ率，3 ズレ率，4 ズレ率すべてを指す．正答率を上げることはカード交換後の手札からより正確に順位予測することであり，それは 1 位となるようなカード交換の解明につながる．そして，一致率を学習回数を変化させて確認した．正答率 (0 ズレ率ともいえる) のほかに 1 ズレ率，2 ズレ率，3 ズレ率，4 ズレ率を調査したのは，誤答の場合に，正答の順位により近い順位を予測するのかを確認するためである．

なお，学習に使用したカード交換後パターン数は 500,000 で，テストに使用したカード交換後パターンは 10,000 個である．

カード交換後実験における学習回数の変化に対する平均二乗誤差と一致率の推移について図 6 に示す．横軸の学習回数は N 回目の学習直前を意味する．右縦軸の平均二乗誤差は学習パターンとの平均二乗誤差である．左縦軸の正答率は R_c と R_d が一致した割合，N ズレ率は実際の順位が R_d のときに予測した順位が $R_c - N$ または $R_c + N$ である割合である．

平均二乗誤差は減少し続け，正答率はほぼ収束し，約 42%となった．誤答の場合，1 ズレ率が最も高い約 47%，4 ズレ率が最も低い約 0%となった．正答率と 1 ズレ率を合わせると約 89%なので，かなり高い確率で順位予測できることがわかる．本実験では単純な三層ニューラルネットワークを利用したが，

次に，学習回数を 1000 回に固定し，学習パターン数を変化させて正答率を確認した．これは学習パターン数が十分かどうかを確認するためである．

カード交換後実験における学習パターン数の変化に対する正答率の推移について図 7 に示す．横軸は学習に使用した学習パターン数である．実際に実験で使用した学習パターン数は 1，5，50，500，5,000，50,000，500,000 の 7 通りである．縦軸の正答率は「学習回数

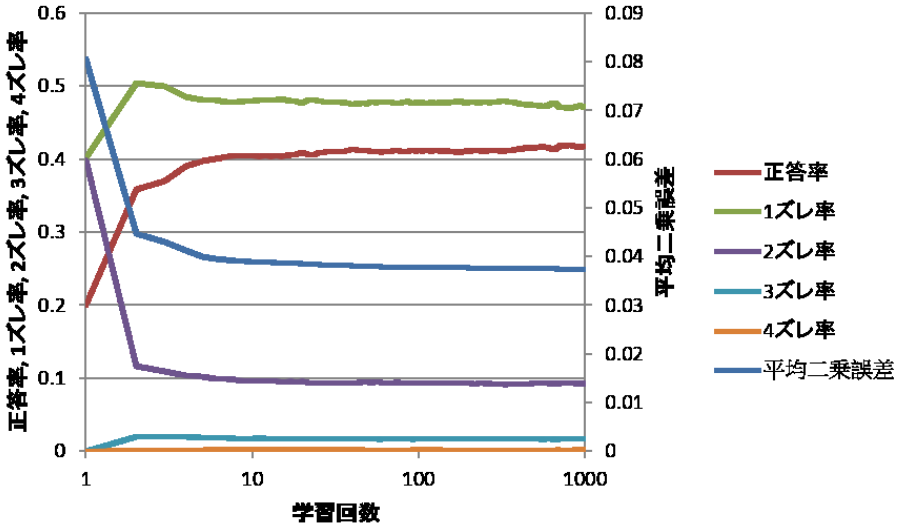

図 6: カード交換後実験における学習回数の変化に対する平均二乗誤差と一致率の推移

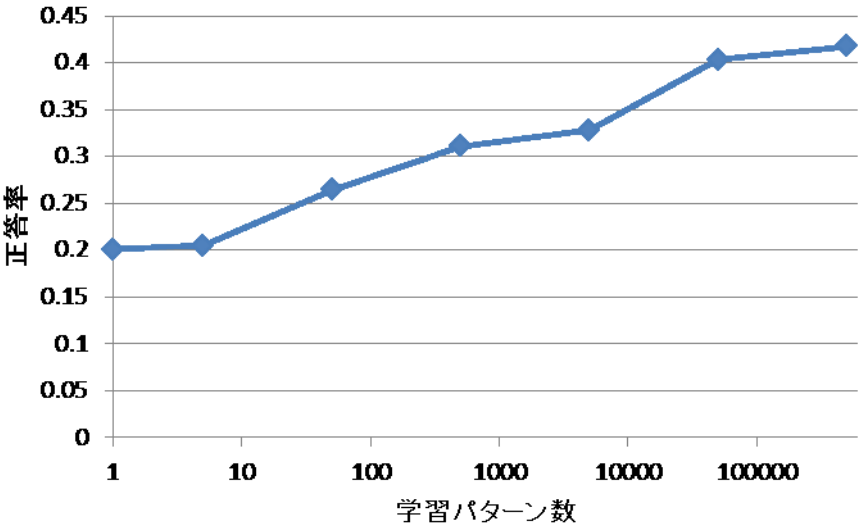

図 7: カード交換後実験における学習パターン数の変化に対する正答率の推移

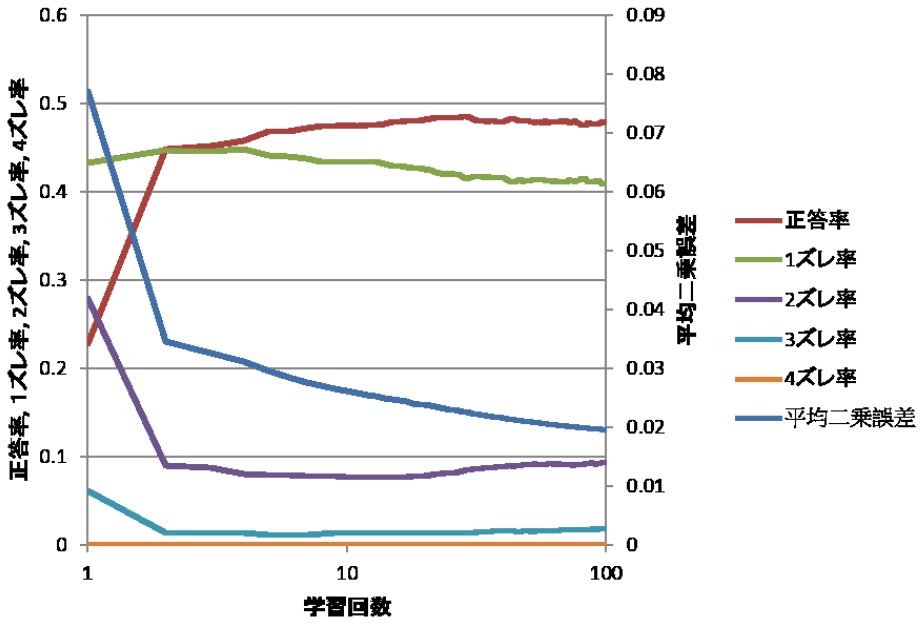

図 8: カード提出後実験における学習回数の変化に対する平均二乗誤差と一致率の推移

の変化に対する平均二乗誤差と一致率の推移」における正答率と同様である．学習パターン数に対して正答率が上昇傾向にあるので，学習パターン数は 500,000 でも不十分だということがわかった．学習パターン数を増やせば，正答率はさらに上昇する可能性がある．

4.7 カード提出後の事後状態から順位予測する実験

順位 R_d で上がることができたカード提出後の事後状態から順位予測を行い，予測した順位 R_c と R_d の一致率を調査した．なお，一致率は後述する正答率，1 ズレ率，2 ズレ率，3 ズレ率，4 ズレ率すべてを指す．正答率を上げることはカード提出後の事後状態からより正確に順位予測することにつながり，それは 1 位となるようなカード提出の解明につながる．そして，一致率を学習回数を変化させて確認した．正答率 (0 ズレ率ともいえる) のほかに 1 ズレ率，2 ズレ率，3 ズレ率，4 ズレ率を調査したのは，誤答の場合に，正答の順位により近い順位を予測するのかを確認するためである．

学習に使用したカード提出後パターン数は 500,000 で，テストに使用したカード提出後パターン数は 10,000 である．

カード提出後実験における学習回数の変化に対する平均二乗誤差と一致率の推移について図 8 に示す．横軸の学習回数は N 回目の学習直前を意味する．右縦軸の平均二乗誤差は学習パターンとの平均二乗誤差である．左縦軸の正答率は R_c と R_d が一致した割合，N ズレ率は実際の順位が R_d のときに予測した順位が $R_c - N$ または $R_c + N$ である割合である．

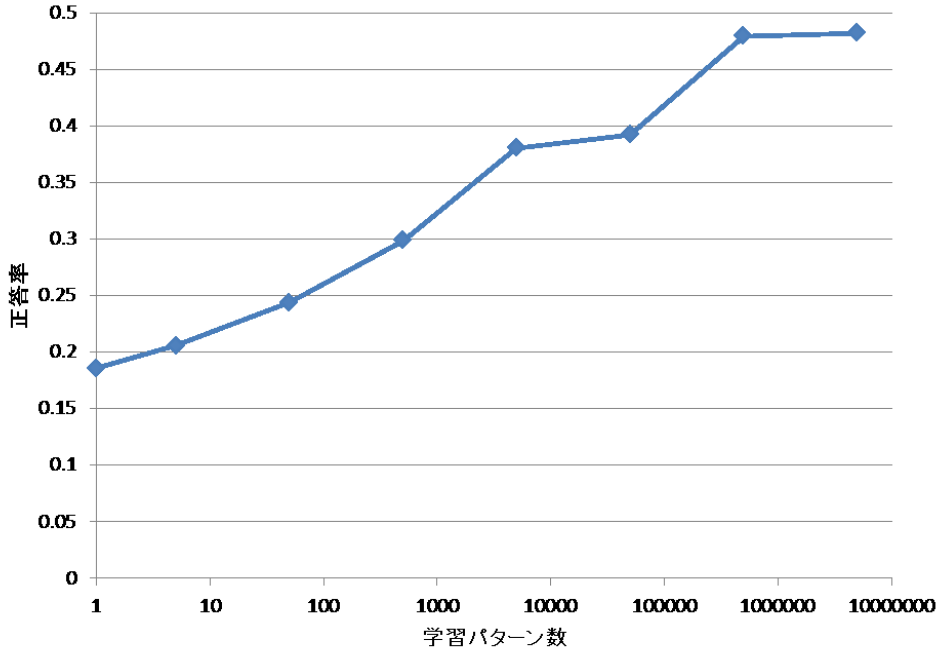

図 9: カード提出後実験における学習パターン数の変化に対する正答率の推移

平均二乗誤差は減少し続け，正答率はほぼ収束し，約 48%となった．誤答の場合，1 ズレ率が最も高い約 41%，4 ズレ率が最も低い約 0%となった．正答率と 1 ズレ率を合わせると約 89%なので，かなり高い確率で順位予測できることがわかる．本実験では単純な三層ニューラルネットワークを利用したが，中間層のユニット数を増やしたり，中間層数を増やしたりすれば，正答率はさらに上昇する可能性がある．

次に，学習回数を 1000 回に固定し，学習パターン数を変化させて正答率を確認した．これは学習パターン数が十分かどうかを確認するためである．

カード提出後実験における学習パターン数の変化に対する正答率の推移について図 9 に示す．横軸は学習に使用した学習パターン数である．実際に実験で使用した学習パターン数は 1，5，50，500，5,000，50,000，500,000 の 7 通りである．縦軸の正答率は学習回数の変化に対する平均二乗誤差と一致率の推移における正答率と同様である．学習パターン数に対して正答率が上昇傾向にあるので，学習パターン数は 500,000 でも不十分だということがわかった．学習パターン数を増やせば，正答率はさらに上昇する可能性がある．

4.8 各特徴の影響を順位予測の正答率で確認する実験

検証する特徴は「カード提出後の事後状態から順位予測する実験」で使用した全特徴，各特徴，特徴なしの 8 種類である．「カード提出後の事後状態から順位予測する実験」で使用したカード提出後パターンのうち，検証する特徴のビットはそのままにしてそのほかのビットはすべて 0 に固定して棋譜データからパターンを作成した．このような設定で

「カード提出後の事後状態から順位予測する実験」と同様に正答率を確認した．この実験をする目的は各特徴が正答率にどの程度影響を与えるのかを確認するためである．

実験に使用した学習パターン数は 500,000，テストパターン数は 10,000 である．

表 3: 各特徴の正答率に対する影響

正答率	特徴
0.48	全特徴
0.40	カード残り枚数のみ
0.34	自分の手札のみ
0.33	相手の手札のみ
0.26	使われたカードのみ
0.23	場に出てるカードのみ
0.23	革命・縛りのみ
0.19	特徴なし

各特徴の正答率に対する影響を表 3 に示す．カード残り枚数が正答率に最も強い影響を与えていると考えられる．ランダムに順位予測する場合，1 位から 5 位までの 5 つの順位から 1 つの順位を選択するので，正答率は 0.2 となるが，特徴なしの場合に正答率が約 0.2 となっているので，つじつまが合っていることを確認できる．さらに，単独で特徴を使用するよりも，特徴を組み合わせた方が正答率が高くなることがわかった．

4.9 カード残り枚数で順位予測する実験

「各特徴の影響を順位予測の正答率で確認する実験」でカード残り枚数が順位予測に最も強い影響を与える特徴だということがわかった．そこで，ニューラルネットワークが，カード残り枚数と順位の関係をどのように認識しているのかを解析した．

この実験ではカード残り枚数の構成についてさらに詳しく 2 通りの方法で考えた．

1 つ目の方法は自分以外の 4 人のプレイヤのカード残り枚数を同じ枚数で固定する場合である．自分のカード残り枚数が 0 枚から 11 枚のそれぞれの場合に対して，各相手のカード残り枚数を 1 枚，2 枚，4 枚，8 枚で固定した．カード残り枚数だけに着目した解析を行いたいので，学習パターンの入力とテストパターンの入力で，カード残り枚数以外をすべて 0 に固定した．このようなテストパターンは自作した．学習パターンは棋譜データからカード残り枚数以外を 0 に固定し作成した．その 500,000 個の学習パターンで 1000 回学習し，ニューラルネットワークの重みや閾値を調整した．

2 つ目の方法はニューラルネットワークに「カード提出後の事後状態から順位予測する実験」で使用した全特徴を入力した場合である．自分以外の 4 人のプレイヤのうちで上がりの人数ごと評価値を求めた．この方法は 1 つ目の方法に比べてより実際的である．なぜなら，この場合のカード残り枚数の構成は実際のゲームで出現したカードの残り枚数の構

成であり，カード残り枚数以外の特徴のビットもそのまま入力しているからである．学習パターンとテストパターンの内容および個数は「カード提出後の事後状態から順位予測する実験」と同様である．

1つ目の方法では，自分のカード残り枚数が少なくなると評価値がどう推移するのかを確認する．また，自分のカード残り枚数と自分以外のプレイヤ間のカード枚数差が評価値にどのような影響を与えるのかを確認する．また，自分のカード残り枚数が0で，自分以外のプレイヤのカード残り枚数が0でないときに，評価値が0.8から1.0までの値になるかどうか，言い換えれば，大富豪だと認識できるかどうかを確認する．

2つ目の方法では，カード残り枚数以外の特徴のビットを0に固定せず，実際の棋譜データから作成したビットを割り当てたときに，それらのビットの影響で評価値がどう変化するのかを確認する．

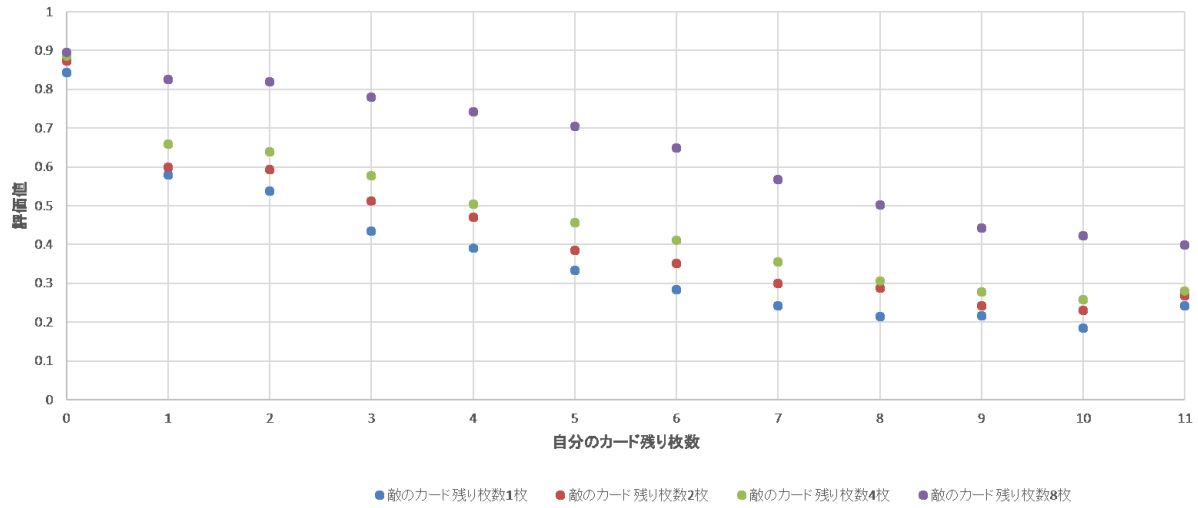

図 10: 相手4人のカード残り枚数を固定した場合における自分のカード残り枚数と評価値の関係

図10に相手4人のカード残り枚数を固定した場合における自分のカード残り枚数と評価値の関係を示す．ニューラルネットワークが，自分のカード残り枚数が少ないほど順位がよくなるという傾向を認識しているいることが確認された．また，自分のカード残り枚数と自分以外のプレイヤ間のカード枚数差が大きいほど，自分がよりよい順位になることも確認できた．また，自分のカード残り枚数が0で，自分以外のプレイヤのカード残り枚数が0でないときに，大富豪だと認識できることも確認できた．また，全員のカード残り枚数が4枚のときと8枚のときは自分の予測順位が3位であった．また，全員のカード残り枚数が1枚のときと2枚のときは自分の予測順位が2位にかぎりなく近い3位であった．

図11から図14に「カード提出後の事後状態から順位予測する実験」で使用した全特徴を入力したときの，自分のカード残り枚数と評価値の関係を示す．

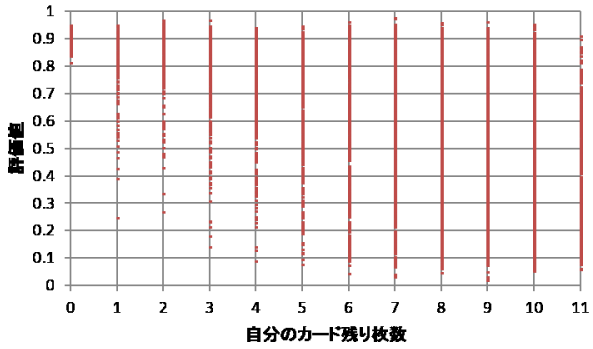

図 11: 相手 4 人のうち上がりの人数が 0 人

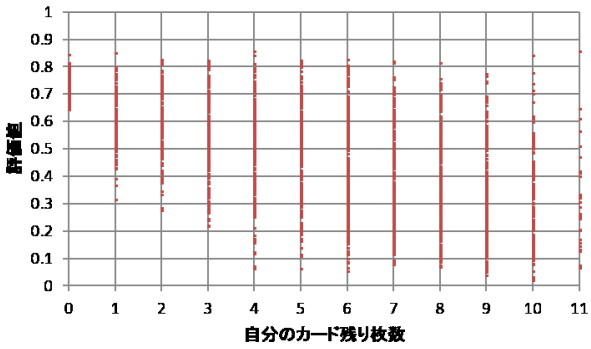

図 12: 相手 4 人のうち上がりの人数が 1 人

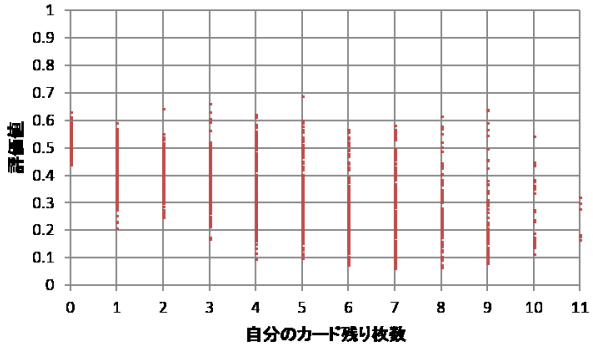

図 13: 相手 4 人のうち上がりの人数が 2 人

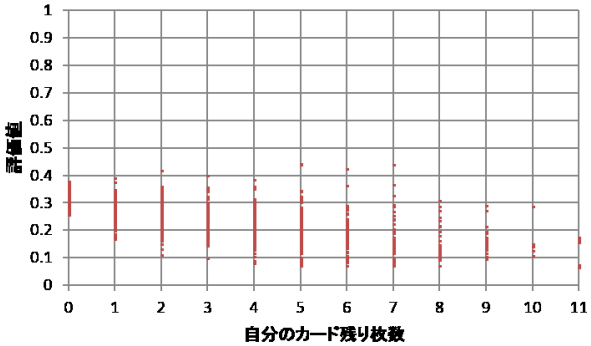

図 14: 相手 4 人のうち上がりの人数が 3 人

図 10 と図 11 を比較すると，図 11 の方が図 10 よりも各自分のカード残り枚数に対して，評価値がばらついていることがわかる．これはカード残り枚数以外の特徴も使うと，事後状態をより精度よく評価できているといえる．

どの場合においても，自分のカード残り枚数が 0 のときは正確な評価値が得られている．すなわち，自分のカード残り枚数が 0 で自分以外に上がりのプレイヤが 0 人の場合は大富豪，自分のカード残り枚数が 0 で自分以外に上がりのプレイヤが 1 人の場合は富豪，自分のカード残り枚数が 0 で自分以外に上がりのプレイヤが 2 人の場合は平民，自分のカード残り枚数が 0 で自分以外に上がりのプレイヤが 3 人の場合は貧民となっている．

さらに，上がりのプレイヤの人数が増えるほど，各自分のカード残り枚数における評価値の幅が小さくなっている．これはプレイヤが上がることで，なることのできない階級が発生するからである．

5　おわりに

5.1　まとめ

本研究では,（大）富豪と（大）貧民間のカード交換後の手札およびカード提出後のゲーム状態から，大富豪から大貧民までの順位予測を試みた．カード交換後の手札から順位予測する実験では，学習パターン数を 50 万まで増やしたが正答率は 42%を越え，なお上昇し続けた．一方，カード提出後の事後状態から順位予測する実験では，学習パターン数を 50 万まで増やしたが正答率は 48%を越え，なお上昇し続けた．各特徴の影響を順位予測の正答率で確認する実験では，順位予測に影響を与えるゲーム状態の特徴は影響力の強いものから順に，カード残り枚数，相手の手札，自分の手札，使われたカード，革命・しばりの有無，場に出てるカードであった．予測順位に最も強い影響を与えると考えられるカード残り枚数について，自分のカード残り枚数が少なく相手のカード残り枚数が多いときに予測順位が高くなることを確認した．特に，自分のカード残り枚数が 0 枚で他プレイヤのカード残り枚数が 0 枚でないときには 1 位を正しく予測した．また，本研究の事後状態で使用した全特徴を使用した場合でも，自分のカード残り枚数が 0 のときには，上がりのプレイヤの人数に応じて，正しい階級を予測できた．

5.2　今後の目標

学習パターン数を増やすと，正答率が向上することを確認したので，TD-Gammon のように，膨大な回数の自己対戦から強化学習するコンピュータ大貧民プレイヤの開発を目指したい．

謝辞

本研究を進めるにあたり，多大なるご指導頂いた保木邦仁助教には心より御礼申し上げます．また，ご協力頂いた研究室の皆様に感謝致します．

参考文献

[1] Michael Buro, “The Othello Match of the Year: Takeshi Murakami vs. Logistello”, ICCA Journal, 20.3, (1997).

[2] Campbell, Murray, A. Joseph Hoane Jr and Feng-hsiung Hsu, “Deep blue”, Artificial Intelligence, 134.1, (2002).

[3] 第 3 回将棋電王戦 HUMAN VS COMPUTER — niconico, http://ex.nicovideo.jp/denou/3rd/match.html visited on 2015/01/06.

[4] 第 2 回電聖戦, http://entcog.c.ooco.jp/entcog/densei/denseisen-2nd.html visited on 2015/01/06.

[5] Stuart Russell, Peter Norvig, *Artificial Intelligence: A Modern Approach*, Pearson, (2010).

[6] 西野哲朗, 不完全情報ゲーム, 情報処理, 53.2, (2012): 112-117.

[7] 地曳隆将, 松崎公紀, “大貧民における棋譜データからの提出手役評価関数の学習”, 情報処理学会研究報告 31 回ゲーム情報学研究会資料集, (2014).

[8] Richard S. Sutton, Andrew G. Barto 原著, 三上貞芳, 皆川雅章 共訳, 強化学習, 森北出版, (2000).

[9] UECda-2014 - UEC コンピュータ大貧民大会, http://uecda.nishino-lab.jp/2014/index.php visited on 2015/02/11.

[10] Gerald Tesauro, “Temporal difference learning and TD-Gammon”, Communications of the ACM 38.3, (1995).

[11] 平野 廣美, C でつくるニューラルネットワーク, パーソナルメディア, (1991).

[12] ダウンロード | UECda2011, http://uecda.nishino-lab.jp/2011/DOWNLOAD.html visited on 2015/02/13.